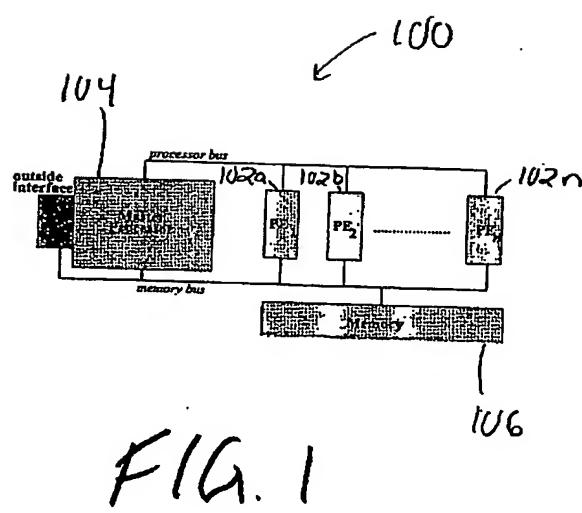


1/12



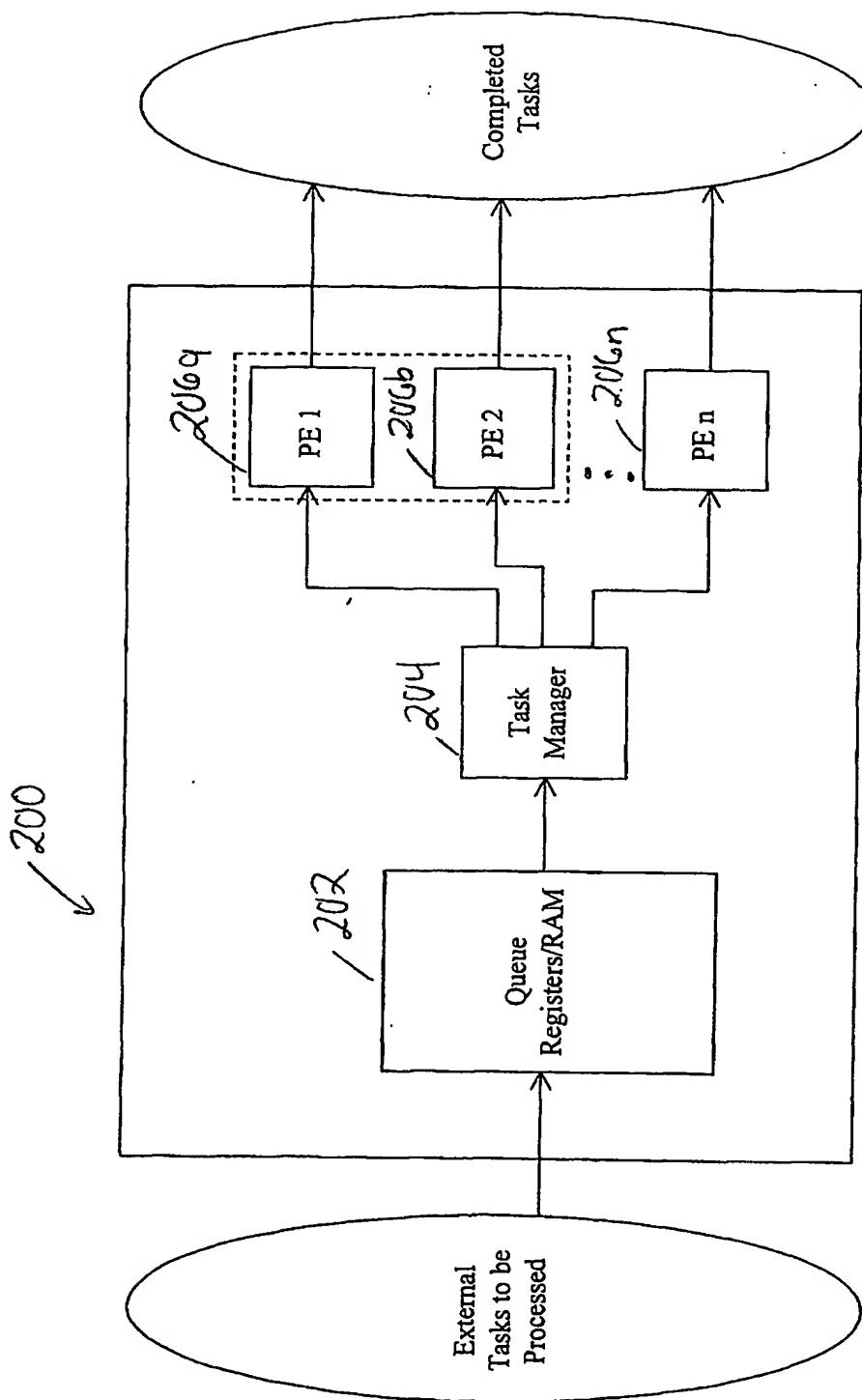
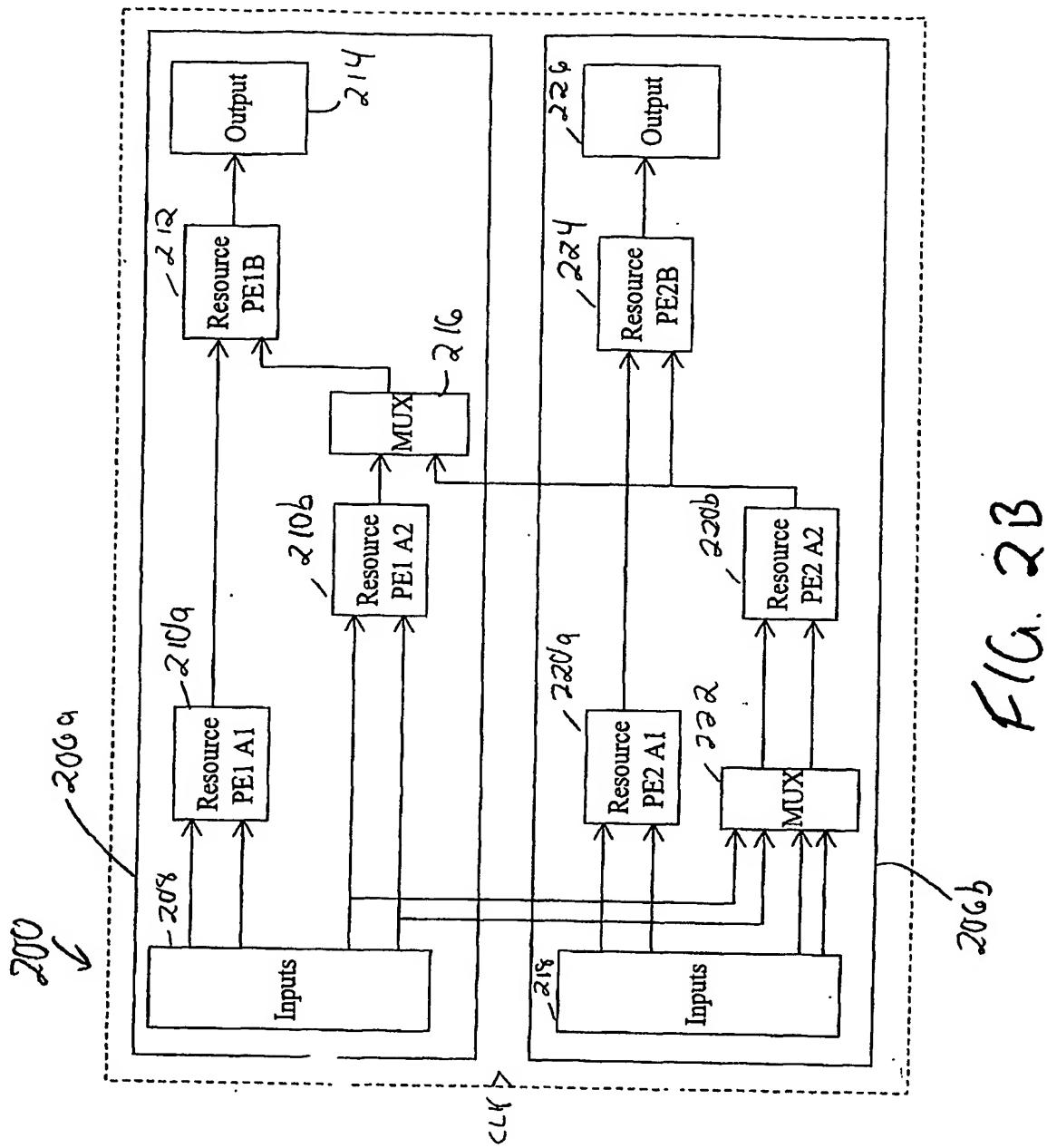
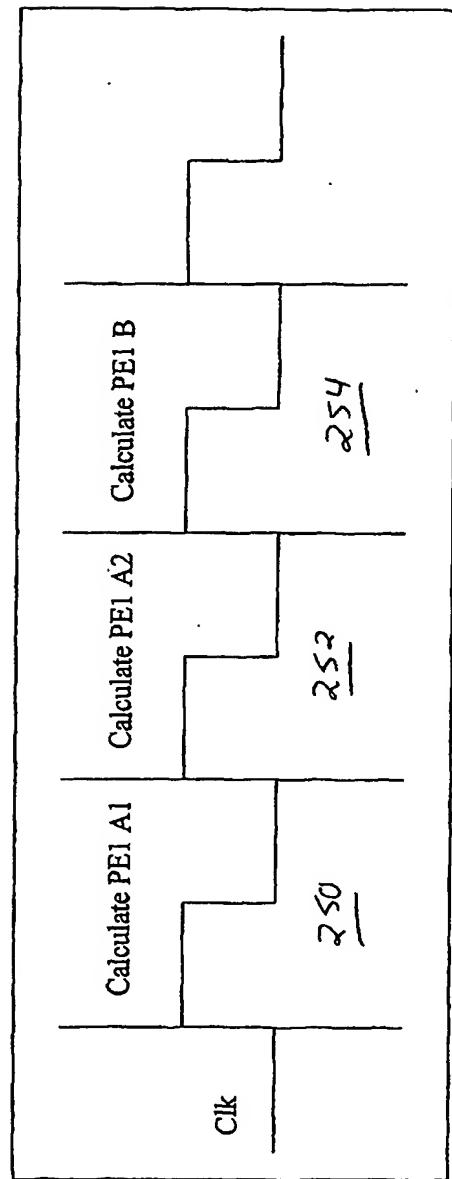


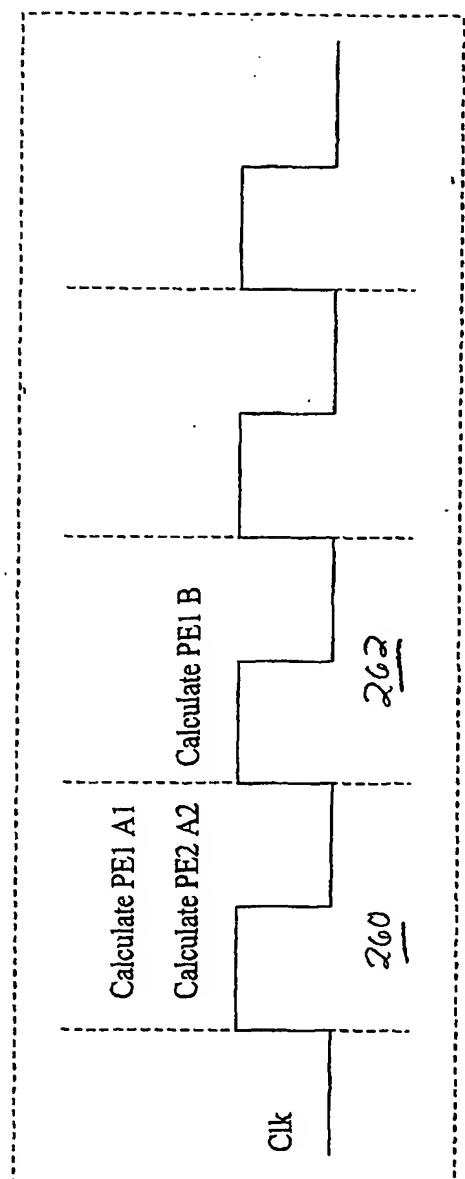
FIG. 2A





Non-threaded Execution

Fig. 2C



Hardware Threaded Execution

Fig. 2D

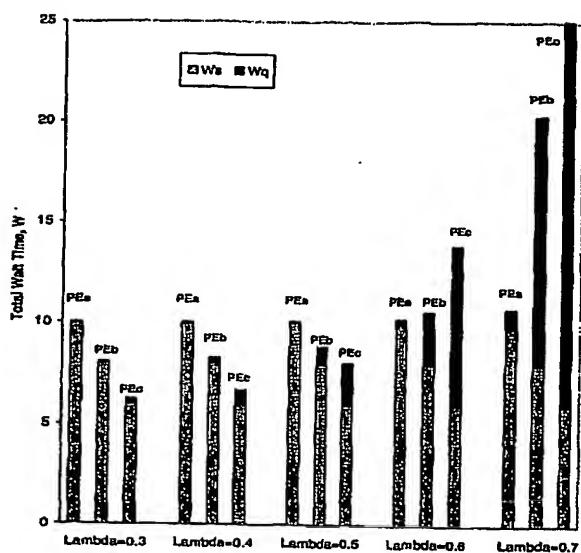


FIG. 3

6/12

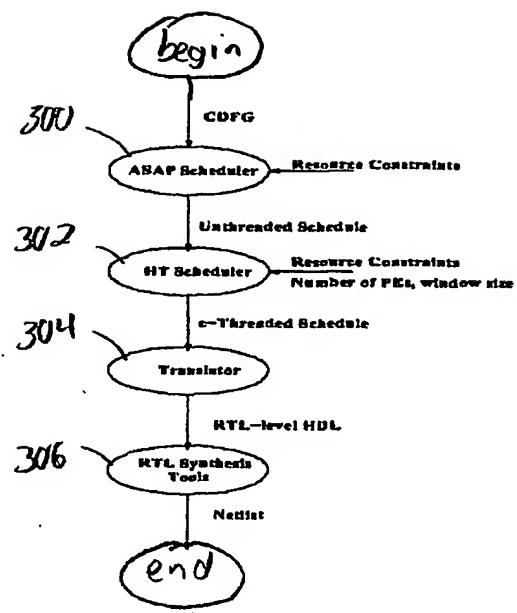


FIG. 4

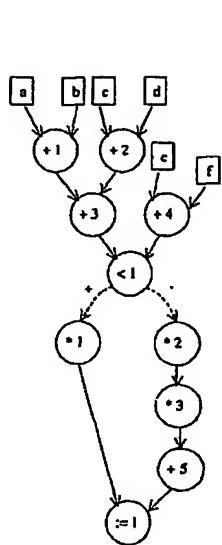


FIG. 5A

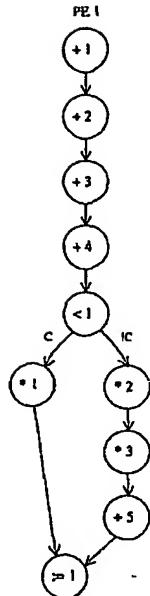


FIG. 5B

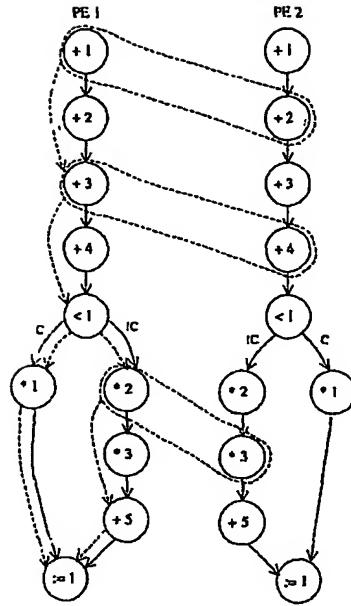


FIG. 5C

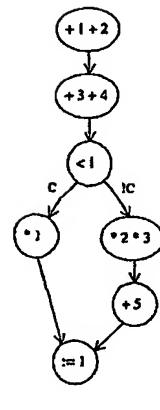


FIG. 5D

| <b>HTscheduler (<math>S, R, n, w</math>)</b>   |  |
|--|--|
| <b>Input:</b>  | $S \equiv$ Unthreaded Schedule<br>$R \equiv$ Resource Constraints Per Partition<br>$n \equiv$ Number of Partitions<br>$w \equiv$ Window Size   |
| <b>Vars:</b>   | $CurrentState \equiv$ Current State to be Scheduled<br>$NextState \equiv$ Next State to be Scheduled<br>$ByPassedEdges \equiv$ Queue of Edges to Bypass<br>$Borrowed \equiv$ Queue of Borrowed States<br>$StatesAlreadyScheduled \equiv$ Array of states already scheduled |
| <b>Output:</b>   | $HTS \equiv$ Hardware-threaded Schedule  |
| 1. $CurrentState = initialState(S);$<br>2. <b>while</b> $CurrentState$ is not Null <b>do</b><br>3. <b>if</b> $IsConditional(CurrentState)$ <b>then</b><br>4. <b>for each</b> child of $CurrentState$<br>5. $HTscheduler(Schedule(child), R, n, w)$<br>6. <b>else if</b> $StatesAlreadyScheduled(CurrentState) = 0$ <b>then</b><br>7. $StatesAlreadyScheduled(CurrentState) = 1$<br>8. $NextState = CurrentState$<br>9. <b>for</b> ( $i = 0; i \leq w; i++$ ) <b>do</b><br>10. $NextState = successor(NextState)$<br>11. <b>if</b> $IsConditional(NextState) \vee IsJoining(NextState)$ <b>then</b><br>12. $i = w$<br>13. <b>else</b><br>14. <b>if</b> $IsSchedulable(CurrentState, NextState, R, n)$ <b>then</b><br>15. $ByPassedEdges.append(outedge(NextState))$<br>16. $Borrowed.append(NextState)$<br>17. $StatesAlreadyScheduled(NextState) = 1$<br>18. $CombineStates(CurrentState, NextState)$<br>19. <b>if</b> $successor(CurrentState) = Borrowed.top$ <b>then</b><br>20. $Borrowed.dequeuetop$<br>21. <b>while</b> $!IsEmpty(Borrowed) \wedge$<br>22. $Borrowed.top = (target(ByPassedEdges.top))$ <b>do</b><br>23. $Borrowed.dequeuetop$<br>24. $ByPassedEdges.dequeuetop$<br>25. $CreateEdge(CurrentState, target(ByPassedEdges.top), dashed)$<br>26. $ByPassedEdges.dequeuetop$<br>27. <b>else</b><br>28. $CreateEdge(CurrentState, successor(CurrentState), dashed)$<br>28. $CurrentState = successor(CurrentState)$ |  |

FIG. 6

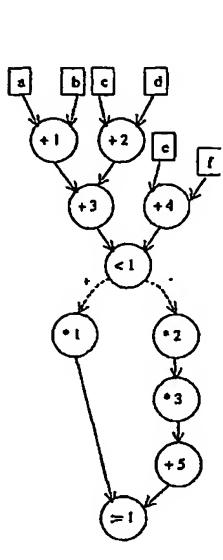


FIG. 7A

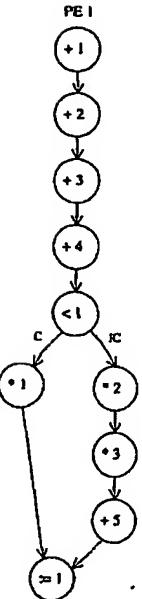


FIG. 7B

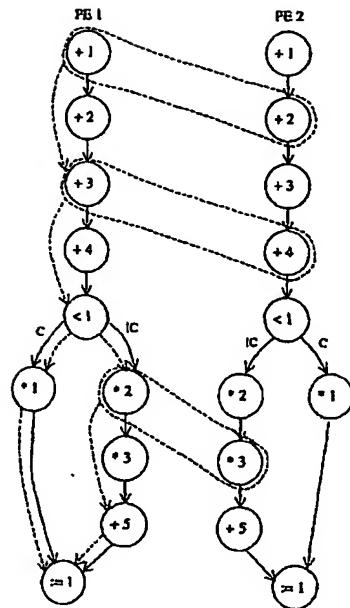


FIG. 7C

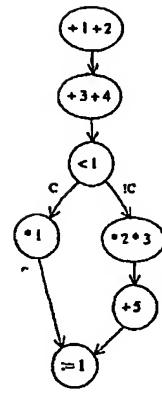


FIG. 7D

10/12

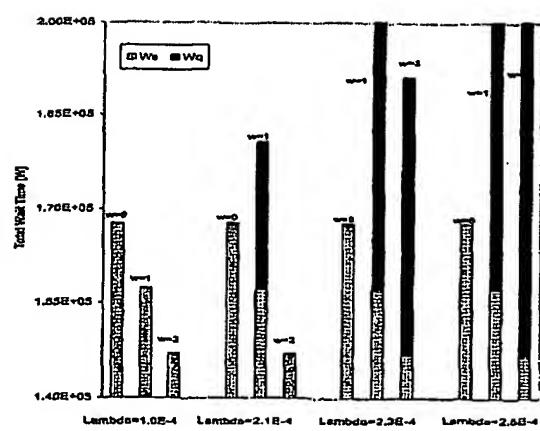


FIG. 8

11/12

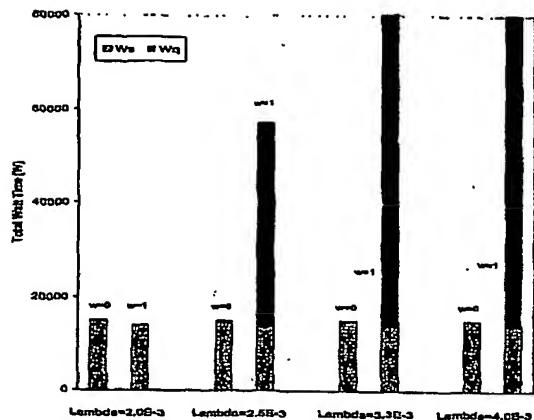


FIG. 9

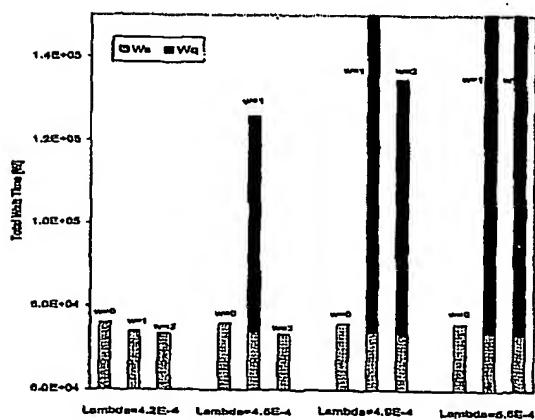


FIG. 10

12/12

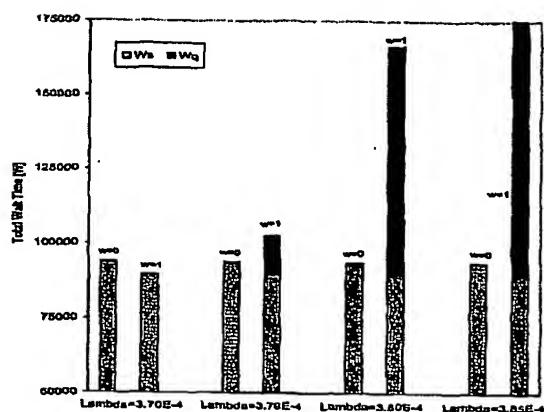


FIG. 11

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**